APPENDIX 1: EXAMPLES OF ALGORITHMIC STEPS

EXAMPLE IMAGE

The Figure A below shows an example image on which some of the algorithmic steps are demonstrated. The size of this image is 20 rows by 20 columns.

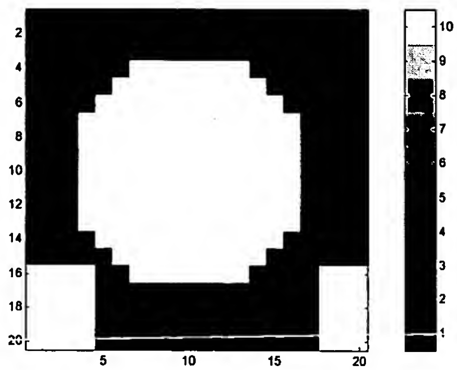


Figure A: Example input image with 20 rows and 20 columns.

The image shown above is represented as a matrix of numbers and can also be displayed in 20 x 20 matrix form where the black pixels on the image have a number zero and white pixels have a number 10 in this case:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	10	10	10	10	10	10	10	0	0	0	0	0	0	0
0	0	0	0	0	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0
0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0
0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0
0	0	0	0	0	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0
10	10	10	10	0	0	10	10	10	10	10	10	10	0	0	0	0	10	10	10
10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10
10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10
10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10
10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10

To illustrate some of our algorithms, we also use a noisy version of this example image. To create this image, we added random values to every pixel of the example image and this noisy image is shown in Figure B.

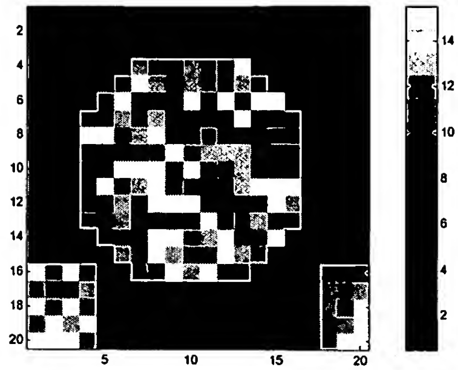


Figure B: Noisy image generated by adding random values to every pixel of the example image shown in Figure A.

As before, the noisy image is shown as a 20 x 20 matrix below:

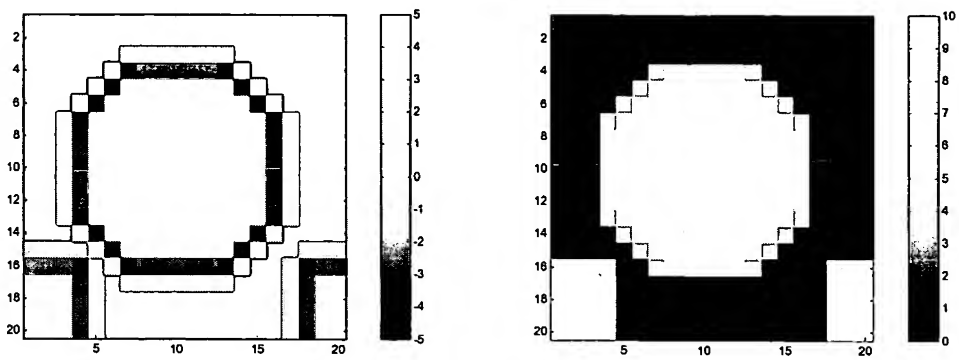
1.4	4.1	1.1	1	3	0.074	2.7	0.37	3.8	0.91	4.5	0.67	3.3	0.52	4.2	2.6	4.6	2.8	0.86	1.7
2	0.83	3.5	2.3	4.7	1.4	3.1	0.97	3.3	2.5	4.7	0.11	4.3	0.79	4	3.6	2.8	3.8	0.65	2.8
2.5	2	2.6	0.41	1.4	4.1	3.4	1.9	4.4	2.1	1.7	1.3	2.8	2	3.5	2.6	3.3	3.9	1.1	0.59
3.6	2.6	4.7	4.3	4.4	4.9	13	11	11	13	12	11	15	2	2.3	3	3.9	2.4	0.53	0.85
1.5	3.6	3.6	2.8	0.51	10	14	14	12	13	12	10	14	10	0.41	4.8	0.53	4	0.71	1.4
0.56	2.8	1.1	1.6	10	14	10	12	11	15	11	14	11	15	14	4.1	0.0054	2.4	2.3	2.8
2.2	2.3	2.2	12	11	13	12	13	10	11	11	11	14	11	11	12	2.7	1	3.9	2.4
2.3	2.2	0.86	14	15	13	14	15	10	11	13	10	11	12	12	13	0.034	2.9	1.4	4.8
0.073	0.44	4.8	12	10	11	12	13	14	13	14	14	13	12	10	11	2.3	3.3	1.1	1.2
3.3	2.2	1.8	10	11	14	15	15	12	15	12	13	13	11	12	11	0.98	3.4	4.5	2.4
3.6	1.8	0.25	11	15	11	13	14	12	10	12	11	14	14	14	14	3.9	4.7	0.037	2.6
1.4	1.5	3.8	10	11	14	10	15	15	14	11	12	12	12	14	13	3.1	3.9	2.9	4
1.3	4.3	4.5	13	12	13	10	11	11	10	14	10	12	14	12	10	0.078	3.7	2.7	0.97
3.5	3.8	1.4	0.61	11	11	10	14	14	13	13	14	12	12	14	2.8	4.5	4.3	3.3	4.5
3.9	4.7	1.3	2.6	3.4	13	11	15	13	11	11	14	13	12	2.4	2.3	3.8	5	1.6	4.6
15	13	15	11	4.8	3	13	11	14	14	15	11	11	2.5	2.8	4.5	4.5	13	11	10
12	10	11	14	3.8	3.3	2.3	0.61	2.4	1.3	3.2	2.8	0.01	0.85	3.1	1.4	3.8	13	12	14
15	13	15	12	3.3	0.92	3.5	3.8	5	4.7	1.1	0.8	4	2.6	3.3	0.33	1.9	14	11	15
12	14	14	14	0.65	3.2	2.9	3.6	1.9	0.69	3.4	3	2.6	3.2	3.1	2.4	1.7	12	13	14
14	15	14	10	0.48	0.85	2.5	3.3	2.7	2.6	3.3	1.7	1.1	0.081	3.4	4.9	2.5	13	15	15

HEAT FILTER – 514

The Laplacian (second derivative) of the example image is computed as the sum of the second partial x-derivative and second partial y-derivative of the image: $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$. This Laplacian is the right-hand side of the heat equation E1.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

The Laplacian of the example image is computed and this is shown in Figure C(1). Notice the negative values on the inside of the bright areas and positive values on the outside of the bright areas at the edges of the regions. The Laplacian is zero an pixels far from edges.

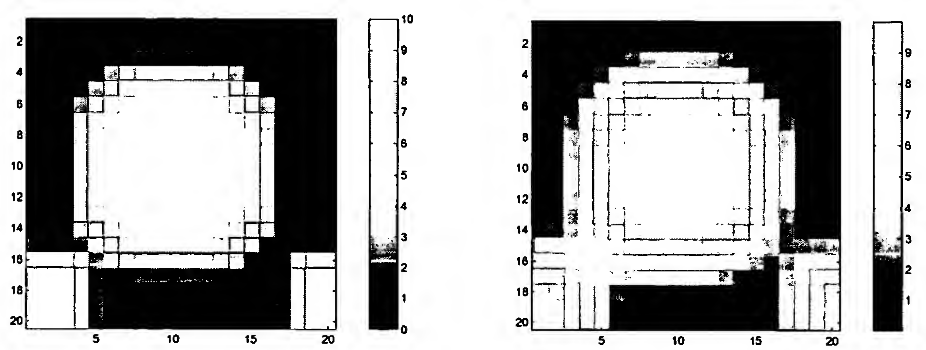


(1) (2)

Figure C (1): The Laplacian of the example image shown in Figure A. (2) The Laplacian added back to the input image.

When this Laplacian is added back to the input image, after multiplying with a step size parameter, it results in blurring of the edges since the bright areas are reduced in brightness and the dark areas are increased in brightness – this output is shown in Figure C(2). This is the output of a single iteration of the heat filter. The heat filter outputs after 4 iterations and after 20 iterations are shown in (1) (2)

Figure D. Notice the progressive blurring of the image as you apply more iterations.



(1) (2)

Figure D: Heat filter outputs after 4 iterations (1) and after 20 iterations (2) on the image shown in Figure A.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

Applying 20 iterations of the heat filter to the noisy image from Figure B results in a reduction of noise and a blurring of the image. This output is shown in Figure E.

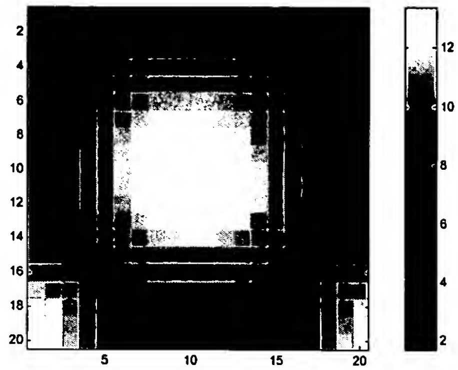


Figure E: The result of applying heat filter to a noisy image.

Note that in this image, the noise has been significantly removed, although the image is very blurred. This image looks very similar to the heat filtered version of the noiseless image shown in Figure D(2).

SHOCK FILTER – 518

The application of the shock filtered to the blurred image is shown in Figure E to sharpen this image. For purposes of the shock filter, the Laplacian of an image (Equation E4) is computed as:

$$\ell(u) = u_{xx}u_x^{\,2} + 2u_{xy}u_xu_y + u_{yy}u_y^{\,2}.$$

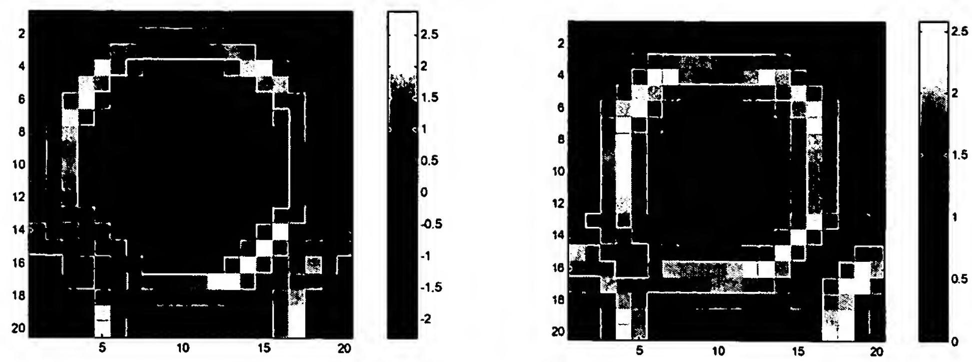
This is computed for the blurred image and is shown below in shown in Figure F(1). Again, notice the negative values on the inside of the bright areas and positive values on the outside of the bright areas at the edges of the regions.

The gradient magnitude of the blurred image (Equation E3) is:

$$\boxed{\|\nabla u\| = \sqrt{u_x^2 + u_y^2}}$$

This is computed for the blurred image and is shown in Figure F(2) below. Note that the gradient magnitude is highest at the edges and zero in smooth regions.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301



(1) (2)

Figure F (1) The Laplacian of the blurred image. (2) The gradient magnitude of the blurred image.

Using a threshold value, t, on pixel gradient to be 0.7 we can compute the F function, which is positive where the Laplacian is greater than zero and the gradient magnitude is greater than a threshold, negative where the Laplacian is less than zero and the gradient magnitude is greater than the threshold and zero elsewhere:

$$
\begin{aligned}
F(\ell(u)) &= 1, \text{ if } \ell(u) > 0 \text{ and} \|\nabla u\| > t \\
&= \text{-}1, \text{ if } \ell(u) < 0 \text{ and} \|\nabla u\| > t \\
&= 0, \text{ otherwise}
\end{aligned}
$$

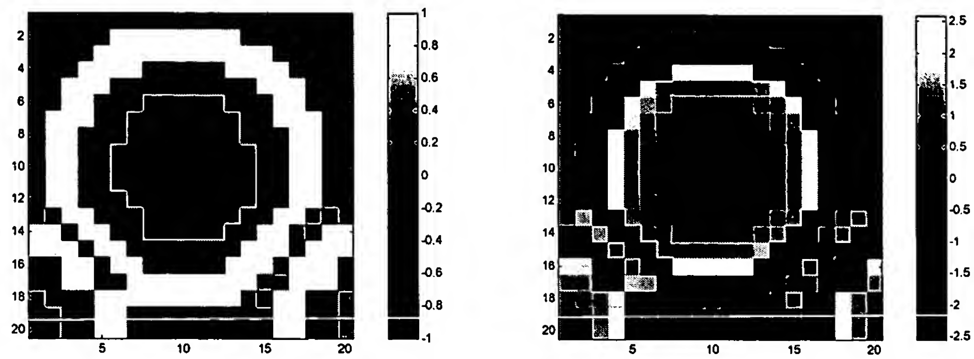
This F image is shown in Figure G(1) below. Note that as with the Laplacian image, everything outside the original white regions at the edge is 1, everything inside the original white regions at the edge is –1. The boundaries of the original region can be seen as the transition between the negative and the positive F values (the zero crossings).

Now, the right hand side of the shock filter equation E2, is the negative of the product of the F function image Figure G(1) and the gradient magnitude image shown in Figure F(2): $-F(\ell(u))\|\nabla u\|$

This right hand side of the shock filter equation is shown in Figure G(2). If we add this image to the original blurred image we can see that it has the opposite effect of the heat filter. The dark pixels in Figure G(2) are subtracted from the blurred image and the bright pixels are added to the blurred image basically, thereby making the blurred image more crisp.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

The addition of this image to the input blurred image is basically what constitutes a single iteration of the shock filter – this addition is shown in Figure H(1). After 20 iterations, a crisper output as shown in Figure H(2) is obtained. While this image is not identical to the input noiseless image of Figure A, it is quite similar to it and much more noise free as compared to the noisy version shown in Figure B.



(1) (2)

Figure G: (1) The F function computed from the Laplacian and the gradient. (2) The negative of the product of the F function and the gradient.

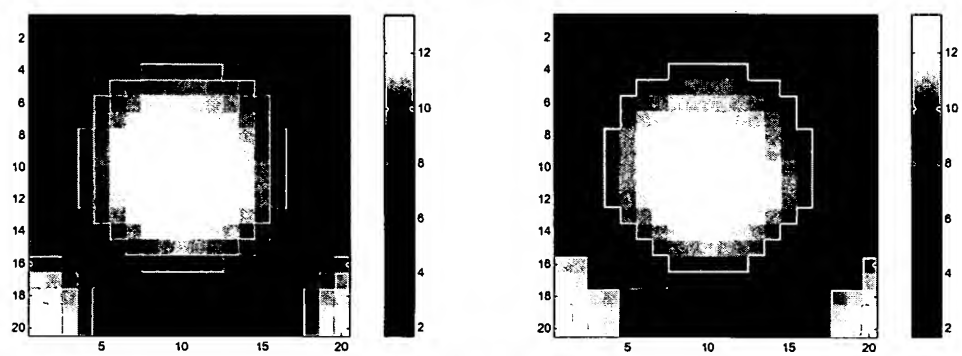


Figure H(1) The output of 1 iteration of the shock filter on the blurred image from Figure E. (2) The output of 20 iterations of the shock filter on the same blurred image.

INTENSITY CLUSTERING – 422

In intensity clustering, the enhanced image is subject to the k-means clustering algorithm. In this example, the enhanced image, shown in Figure H(2), is clustered into 4 clusters.

The minimum and maximum intensity values are 1.69 and 13.40. We divide that range into 4 partitions and get the following initial cluster boundaries.

1.19 4.12 7.04 9.98 13.90

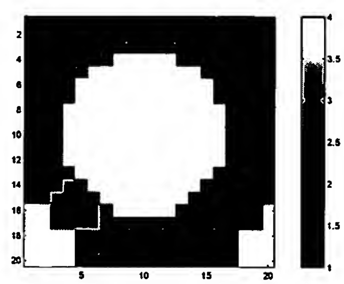
With this partition, all pixels with values between 1.19 and 4.12 are assigned into cluster 1, all between 4.12 and 7.04 are assigned into cluster 2, etc. This initial assignment of every pixel to the 4 clusters is shown in Figure I(1) . Based on this assignment new cluster centers are found and new partition boundaries found accordingly – after this first iteration, the cluster boundaries move slightly and they are now at:

1.19 4.08 6.30 9.57 13.90.

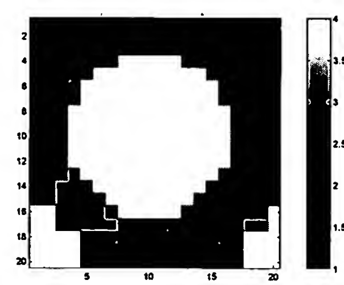
In the next iteration, the pixel assignment based on these partition boundaries is shown in Figure I(2). Finally, after 6 iterations, the partition boundaries change very little between iterations and are set to:

1.19 3.65 5.646 9.25 13.90

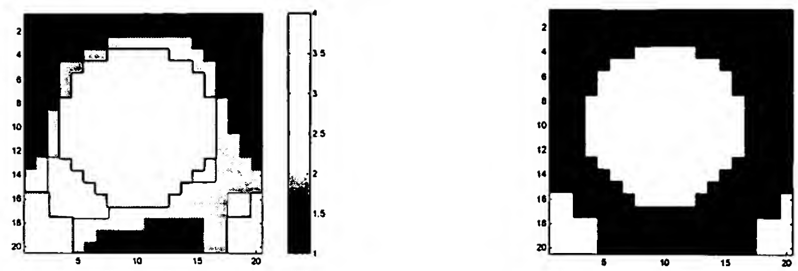
The final assignment of pixels to the 4 clusters is shown in Figure I(3). For this example, the output of the clustering step is the set of pixels assigned to the brightest cluster. This output is shown in Figure I(4).



(1)



(2)

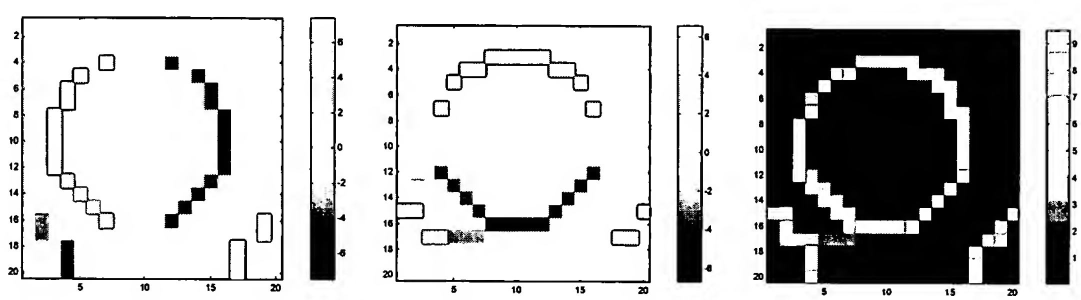


(3) (4)

Figure I: The output of clustering at different iterations of the k-means clustering process. (1) Initial assignment (2) After the first iteration (3) Final assignment after 6 iterations. (4) The one brightest cluster representing the region of interest.

SPATIAL GRADIENT – 526

To compute the gradient magnitude of the enhanced image shown in Figure H(2), the x-derivative u_x and the y-derivative u_y of the image is calculated and then the sum of squares of the two images is calculated to get the gradient magnitude -- $\boxed{\|\nabla u\| = \sqrt{u_x^2 + u_y^2}}$. The x- and y-derivatives and the gradient magnitudes of the enhanced image are shown in Figure J.



(1) (2) (3)

Figure J: Spatial gradients of the image shown in Figure H(2): (1) x-derivative of image (2) y-derivative of image (3) gradient magnitude of image.

HYSTERESIS THRESHOLD – 530

In the hysteresis threshold, two threshold values based on the percentage of non-edge pixels desired and the ratio of the two thresholds is determined. In this example, the percentage non-edge pixels desired is set to 90% and the lower to upper threshold ratio is set to 0.5.

The upper threshold is first computed by looking at the histogram and the cumulative histogram of the gradient magnitude image shown in Figure J(3). The histogram of the gradient magnitude image is shown in Figure K(1). The histogram shows the count of pixels of different gradient values on the image. The cumulative histogram calculated by cumulatively adding up the bins of the histogram is shown in Figure K(2). The cumulative histogram shows the count of all pixels less than a given value in the gradient magnitude image. The horizontal line shown in Figure K(2) corresponds to a y-value of 90% of the image pixels (0.90 * 20 * 20 = 360). The intersection of this line with the cumulative histogram gives the threshold at which 90% of the image pixels will be marked as non-edge and 10% will be marked as edge. From the graph, we can see that a threshold of 7 satisfies that condition. The lower threshold is set to 0.5 times the upper threshold of 7, which gives 3.5.

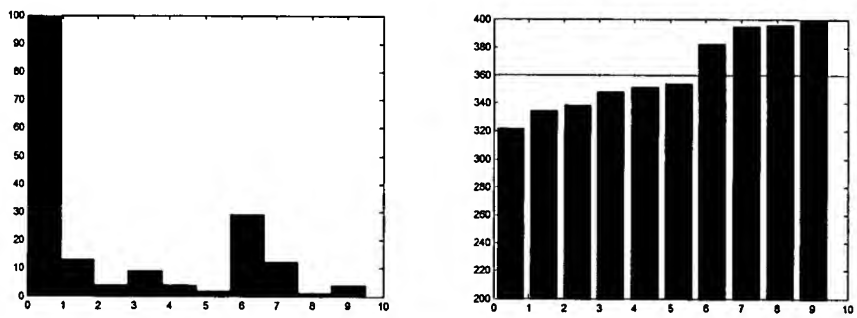
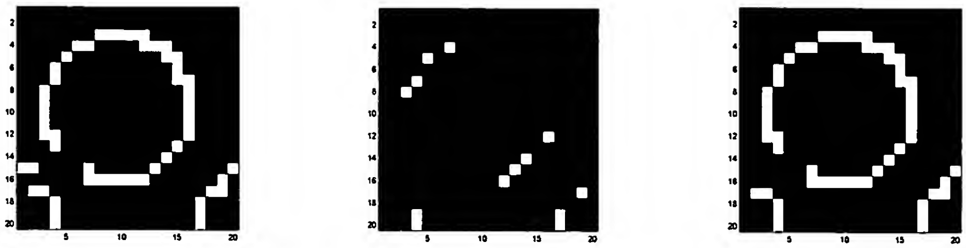


Figure K: Histogram and cumulative histogram of the gradient magnitude image.

For the hysteresis threshold, the threshold of the image at the two threshold values of 3.5 and at 7 is set and then all edge segments from the lower threshold image is selected that have at least one higher threshold pixel. The two thresholded images and the output of the Hysteresis threshold step are shown in Figure L. In this example, note that one edge segment – (row 15, column 2) has no pixel greater than the higher threshold and is therefore removed from the output. All other edge segments from Figure L(1) are retained in the output.

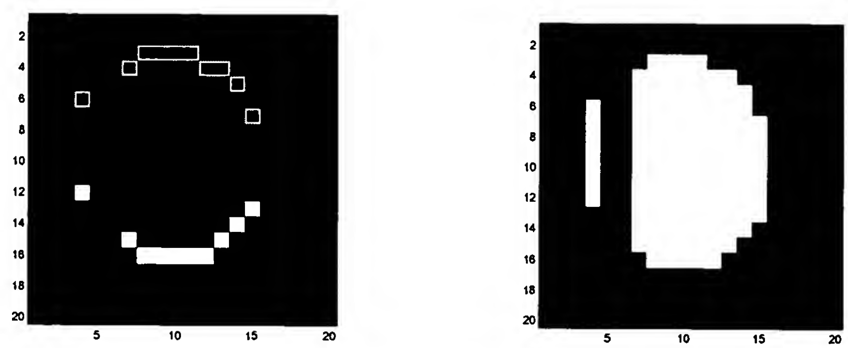


(1) (2) (3)

Figure L: (1) Gradient image thresholded at 3.5 – all pixels greater than 3.5 are shown as white (2) gradient image thresholded at 7, (3) Output of hysteresis threshold.

MATCHING EDGES FILTER – 538

The matching edges filter selects pairs of matching edges from among all the edges found by the hysteresis threshold step (Figure L(3)). The y-derivative image , shown in Figure J(2) is used in this step to help find the vertical matching edges. The matching edges found by this step of the algorithm are shown in Figure M(1) and the filled region between the leading the trailing edges are shown in Figure M(2). Note that this step gets rid of edges of regions on the periphery of the image since it cannot find matching trailing edges for those regions. Also, note that in the output filled image we have some unfilled lines where the gradient was primarily horizontal or where the matching edges did not exist.



(1) (2)

Figure M: (1) Matching edges found by the algorithm – leading edges shown in gray and lagging edges shown in white. (2) The region between the leading the lagging edges filled by white pixels.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

AND IMAGES – 442

The two images shown in Figure I(4) and Figure M(2) represent the segmentation of the input noisy image shown in Figure B. Neither of them represents a perfect output. The region-based algorithm finds extraneous regions while the edge-based algorithm misses part of the desired region. The two images are combined using a Boolean AND operator and the output is shown in Figure N.

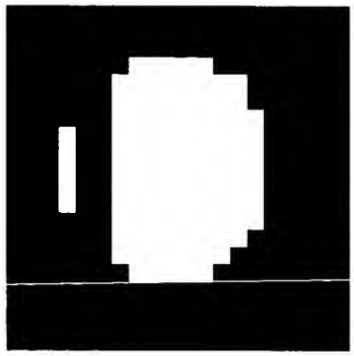


Figure N: Combining the region-based and edge-based segmentation.

CLOSE AND OPEN – 546 & 550

Using a structuring element shaped like a line of width 5 pixels, we close the output of AND images and this gives us a single region representing the structure of interest as shown in Figure O.

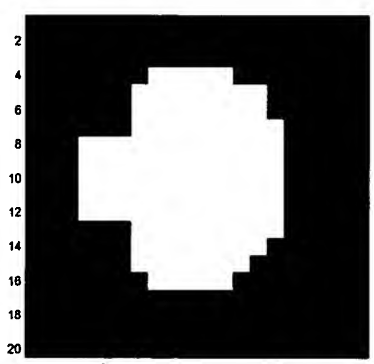
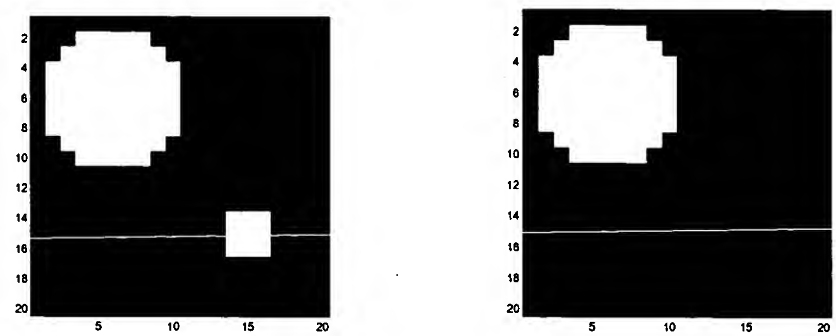


Figure O: Result of closing the image from Figure N.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

Opening this image does not change the result – however, Figure P shows an example of opening on another image. The input image, Figure P(1), contains two regions, one large region and one small region. The large region is 10 pixels in width and height while the small region is 4 pixels in width and height. Using a structuring element shaped like a line of length 5 pixels and applying morphological opening to the input image results in an image shown in Figure P(2) where the smaller region whose width was smaller than the structuring element was removed from the image.



(1) (2)

Figure P: (1) Input image containing two foreground regions. (2) Output image after opening where the smaller region was removed because it was smaller than the structuring element.

FILTER DEEP REGIONS – 554

This filtering step is used to remove regions from a segmented image that start at a location deeper than a user specified depth threshold. This filtering is illustrated in Figure Q, where the input image contains two regions -- one region starts at row 2 and the other region starts at row 14. For each region on the input image, the starting row is determined after doing a connected component labeling – then using a depth threshold of 12, the regions where the starting row depth is greater than the depth threshold are removed from the input image.

BLACK LOWE & GRAHAM PLLC

816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

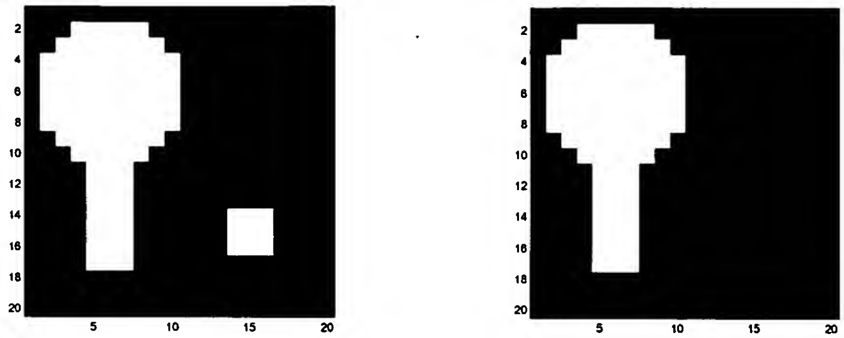


Figure Q: (1) Input image containing two foreground regions. (2) Output image after removing the region whose starting row is greater than the depth threshold of 12.

HEAD DIAMETER RANGE – 730

For a gestational age of 20 weeks, the bi-parietal diameter (BPD) is typically 46 mm. The calculated Head Diameter search range, which ranges from -30% to +30% of the typical BPD value, is found to be:

32.2 36.8 41.4 46.0 50.6 55.2 59.8

For a gestational age of 36 weeks, the bi-parietal diameter (BPD) is typically 88 mm. The calculated Head Diameter search range, which ranges from -30% to +30% of the typical BPD value, is found to be:

61.6 70.4 79.2 88.0 96.8 105.6 114.4

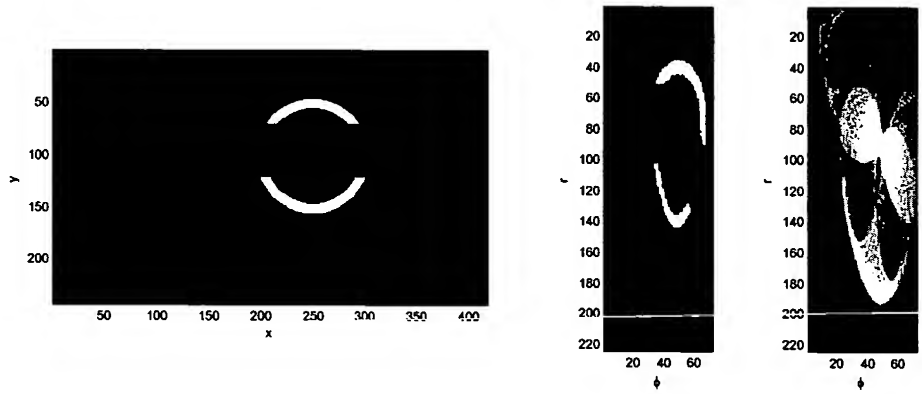
POLAR HOUGH TRANSFORM – 738

To illustrate the polar Hough transform, an exemplary image is generated of a circle in the Cartesian coordinates using the circle equations, E5. This circle image was then dilated to result in a disk like shape and then a few rows of data were set to zero. This input image is shown in Figure R(1). The goal of the circular Hough transform is to find the circle that best fits the white pixels in this input image.

Next, this Cartesian input image was converted to polar domain by inverse scan conversion and the resulting image is shown in Figure R(2). To this polar coordinate image,

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

the polar Hough transform for a circle was applied, where for each white pixel on the image a polar circle of a known radius was drawn. The output of the Hough transform is the sum of all such circles and this is shown in Figure R(3). The center of the best-fit circle has the most points passing through it. So, to determine the best-fit circle, we simply found the pixel on the Hough output with the maximum value. The point with the maximum value represents the center of the best-fit circle.



(1) (2) (3)

Figure R: (1) The input Cartesian coordinate partial circle (2) The partial circle converted to polar coordinates, (3) The Hough transform output showing the sum of polar circles drawn around each white pixel of the input image.

The best-fit circle determined by this Hough transform procedure is overlaid on the polar coordinate image and the Cartesian coordinate image and the two drawings are shown in Figure S.

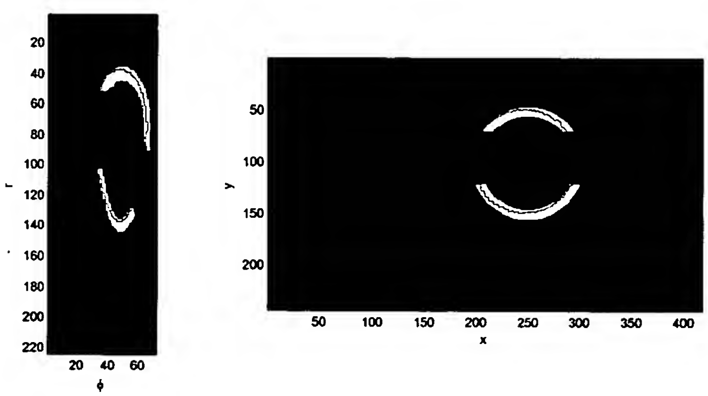


Figure S: The best fit circle found using a Polar coordinate Hough transform (1) overlaid on the polar

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

coordinate input data and (2) overlaid on the Cartesian coordinate input data.

FILL CIRCLE REGION – 746

A circle region in Cartesian space has a somewhat different shape in the Polar coordinate space. To illustrate this polar coordinate circle, we use both Cartesian coordinate and polar coordinate circles. First, a Cartesian coordinate circle of radius 50 pixels and a center (120,250) is drawn using the circle equation, E5, $(x-x_0)^2+(y-y_0)^2=R^2$ -- this circle is shown in Figure T(1).

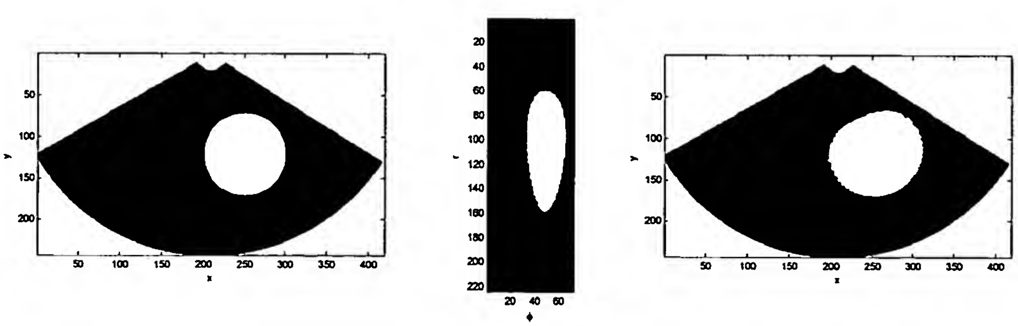
Now, the Cartesian center coordinates (x_0, y_0) are converted from Cartesian to polar center (r_0, ϕ_0) using the scan conversion equations:

$$r_0{}^2=(x_0{}^2+y_0{}^2),$$
$$\phi_0=\tan^{-1}(y_0/x_0).$$

Next, the polar coordinate circle drawing and filling equation, E6, $(r\sin\phi-r_0\sin\phi_0)^2+(r\cos\phi-r_0\cos\phi_0)^2=R^2$,

is applied to draw and fill the same circle of radius R=50, in polar coordinates. This circle in the polar coordinates is shown in Figure T(2).

Finally, to verify that this polar coordinate circle in indeed a representation of the Cartesian coordinate circle, the polar circle image is scan-converted to generate the representative Cartesian coordinate circle image. This scan-converted circle is shown in Figure T(3). Comparing Figure T(1) and Figure T(3), it can be seen that while the two circles are not identical, they are fairly similar. The differences are due to the fact that the polar coordinate data cannot represent Cartesian data completely.



(1) (2) (3)

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301

Figure T Polar coordinate circle filling. (1) A filled circle in a Cartesian coordinate image (2) The equivalent filled circle in polar coordinate image, (3) The scan converted polar coordinate filled circle image.

BLACK LOWE & GRAHAM PLLC
816 Second Avenue
Seattle, Washington 98104
206.381.3300 • F: 206.381.3301